

# **Réalité augmentée Geolocalisation et capteurs**

**Christophe Vestri**

Le mardi 14 mars 2017

# Objectifs du cours

- Connaitre/approfondir la RA
- Avoir quelques bases théoriques
- Expérimenter quelques méthodes et outils
- Réaliser un projet en RA
- Evaluation:
  - Présence (20%)
  - Participation en classe (40%)
  - Projet (40%)

# Plan du cours

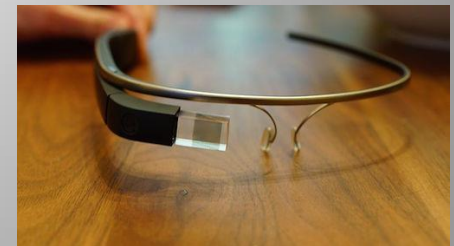
- 28 février : Réalité augmentée intro Html5/JS
- 7 mars: Tag image, Unity/Vuforia projet final
- 14 mars: Leaflet/geoloc/device en JS
- 21 mars: Vision par ordinateur et RA (openCV – C++)
- 28 mars : GeoLoc Unity/Vuforia et Projets

# Plan Cours 3

- Rappel premiers cours
- Cartographie
- Capteurs smartphones
- Géolocalisation et cartes
  - Leaflet
  - Geoloc en Html5
  - Device Events

# RA avec caméra Mobile

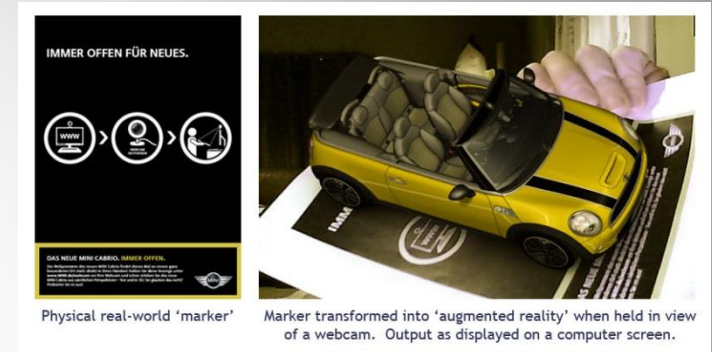
- Smartphones, tout pour la RA
  - Camera + écran – déterminer/montrer ce qui doit être vu
  - Donnée GPS– localisation
  - Compas – quelle direction on regarde
  - Accéléromètre – orientation
  - Connection Internet – fournir des données utiles
- 58% des Français ont un smartphone en 2015
- 90% des 18-24ans
- Lunettes de RA et VR



# Types de RA mobile

## Marqueurs caméras:

- Caméra pour détecter un marqueur dans le monde réel
- Calcul de sa position et orientation
- Augmente la réalité



## Capteurs:

- GPS pour localiser son téléphone
- Recherche de Point d'intérêt proche de nous
- Mesure orientation (compas, accéléromètre)
- Augmente la réalité



# Types de RA mobile

Utilisation de marqueurs caméras:

- Marqueurs Spécifiques:
  - Tag visuels
  - Formes spécifiques (carrés, cercles)
- Marqueurs Images
  - Photo, image de l'objet/scène
- Processus de RA
  - Détection du marqueur dans la vidéo
  - Transformation 2D-3D
  - Affichage 3D





# Applications

- Augmentation de print



IKEA 2014



Idée3com : Application Brisach Vision

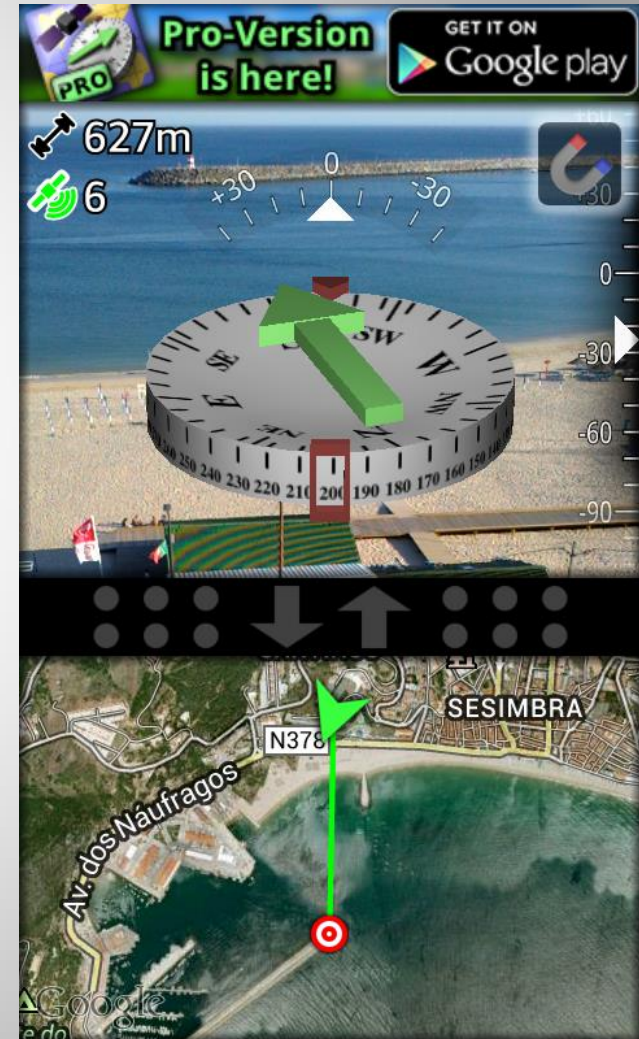




# Types de RA mobile

Utilisation des Capteurs du smartphone:

- GPS pour localiser son téléphone
- Recherche de Point d'intérêt proche de nous
- Mesure orientation (compas, accéléromètre)
- Augmente la réalité



# Applications



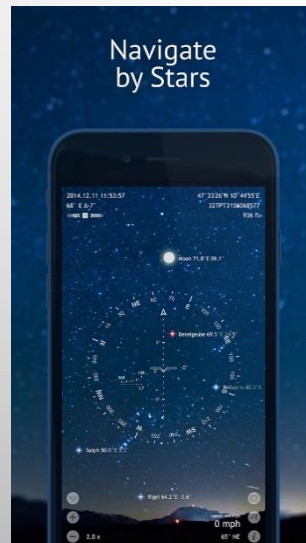
Pokemon Go



Immobilier



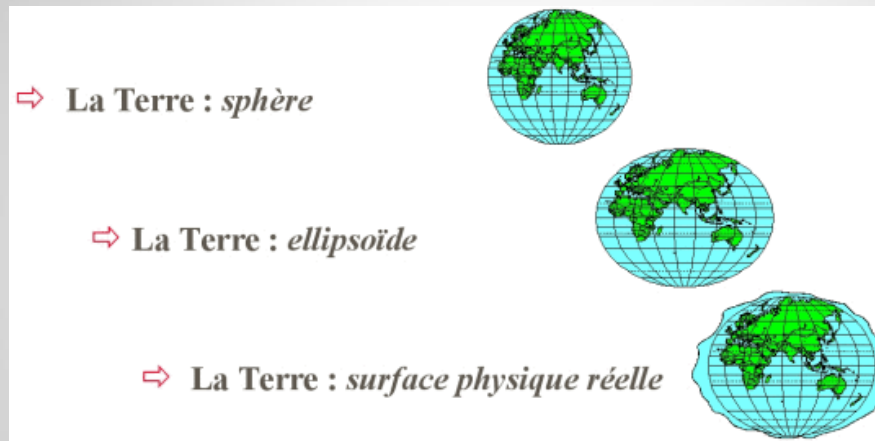
GPS



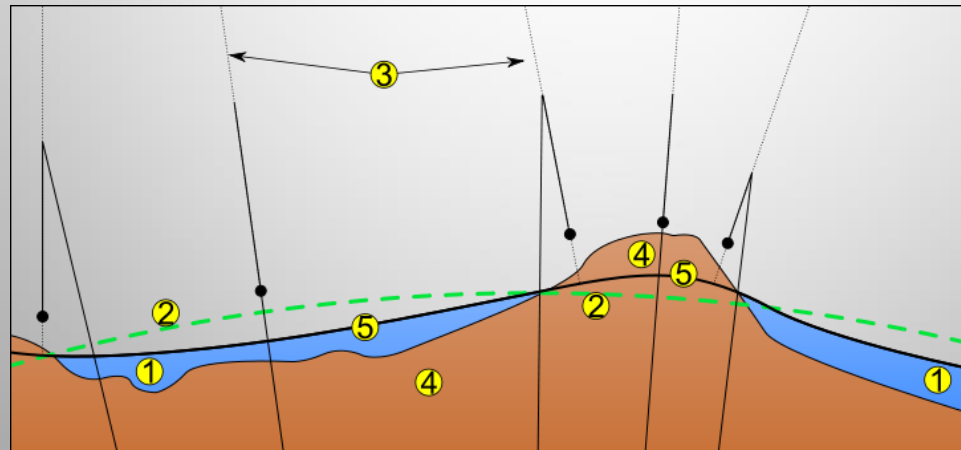
Recherche de points d'intérêts

# Systemes Géographiques et Cartographiques

- Représentation de la terre



Géoïde



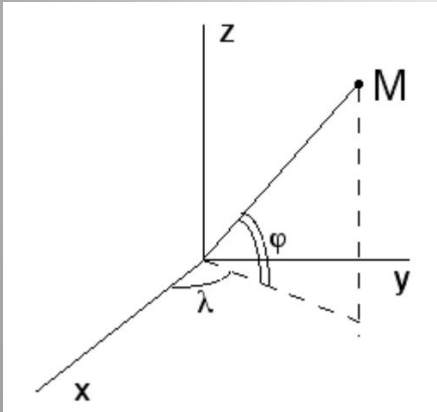
— Ellipsoïde

# Systèmes Géographiques et Cartographiques

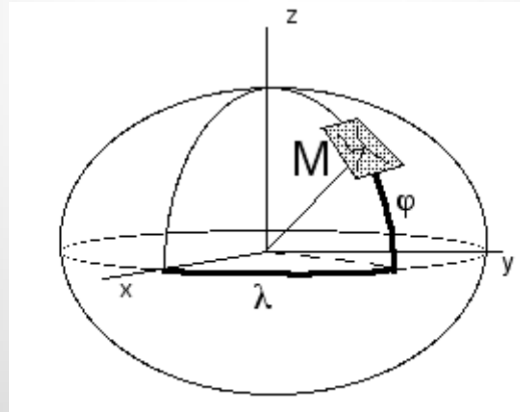
- Construction d'un référentiel géographique

Choix d'un ellipsoïde

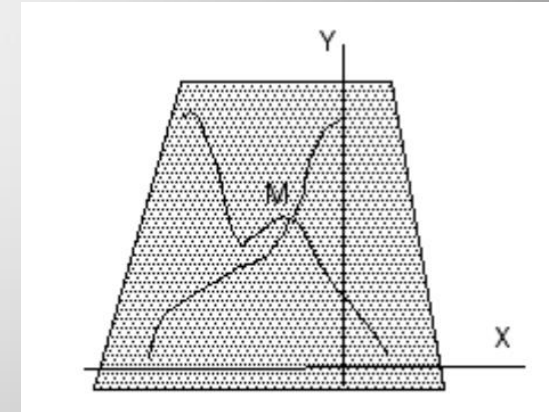
Choix d'une projection



Système de référence  
terrestre (3D)  
 $x, y, z$



Système géographique  
 $\varphi, \lambda$



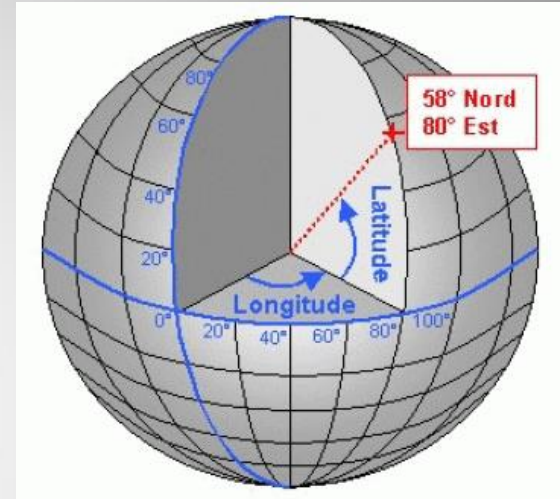
Système cartographique  
 $X, Y$



# Systèmes Géographiques et Cartographiques

- Un point de la surface terrestre est repéré en fonction d'un ellipsoïde par :

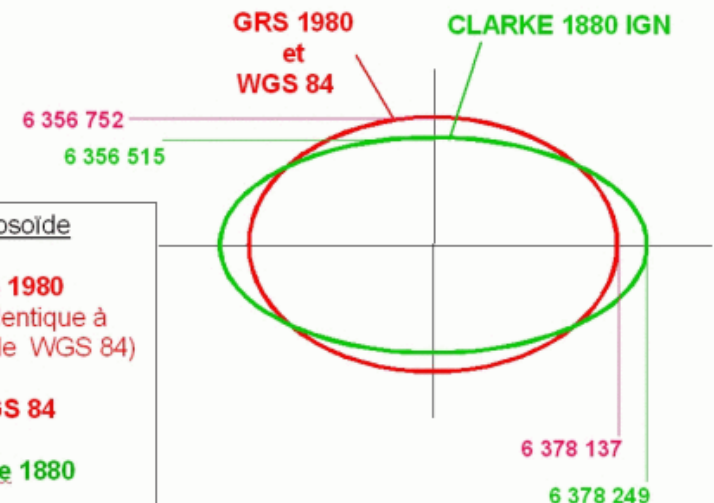
- sa **longitude** :  $\lambda$  (Lambda)
- sa **latitude** :  $\phi$  (Phi)



- Différents systèmes:

- GPS (WGS84),
- France (RGF 93)

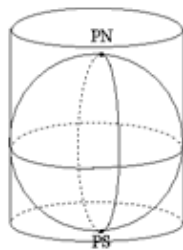
Systèmes	Ellipsoïde
RGF93	<b>GRS 1980</b> (quasi identique à l'ellipsoïde WGS 84)
WGS84	<b>WGS 84</b>
NTF	<u>Clarke 1880</u>



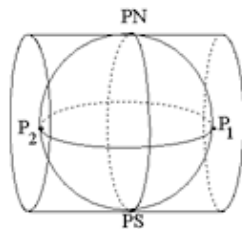
# Systemes Géographiques et Cartographiques

- Choix d'une projection cartographique

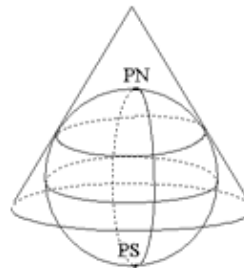
Représentation cylindrique :



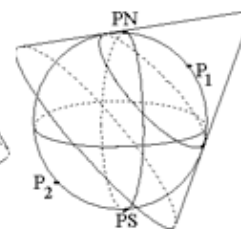
**directe**



**transverse**

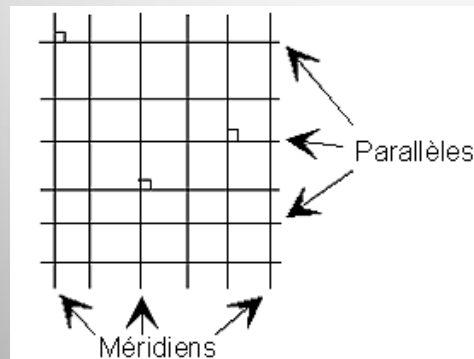
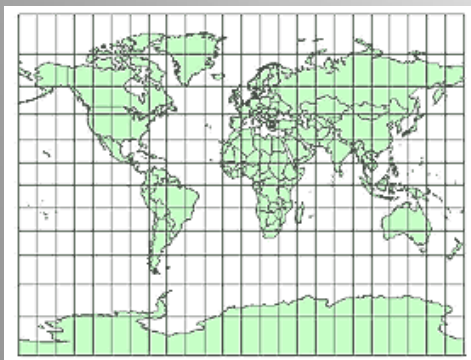
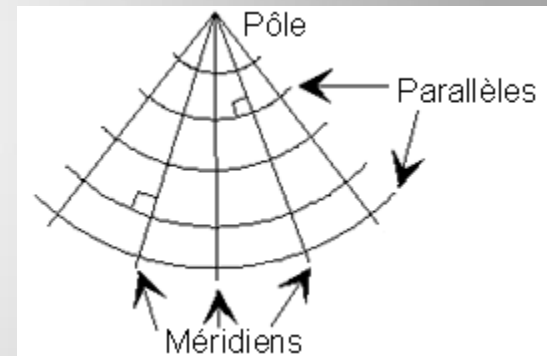


**directe**



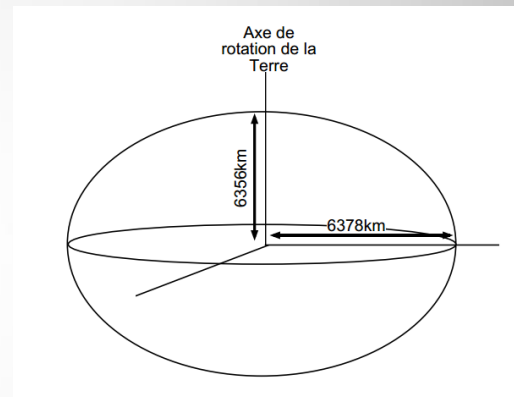
**oblique**

Représentation conique :

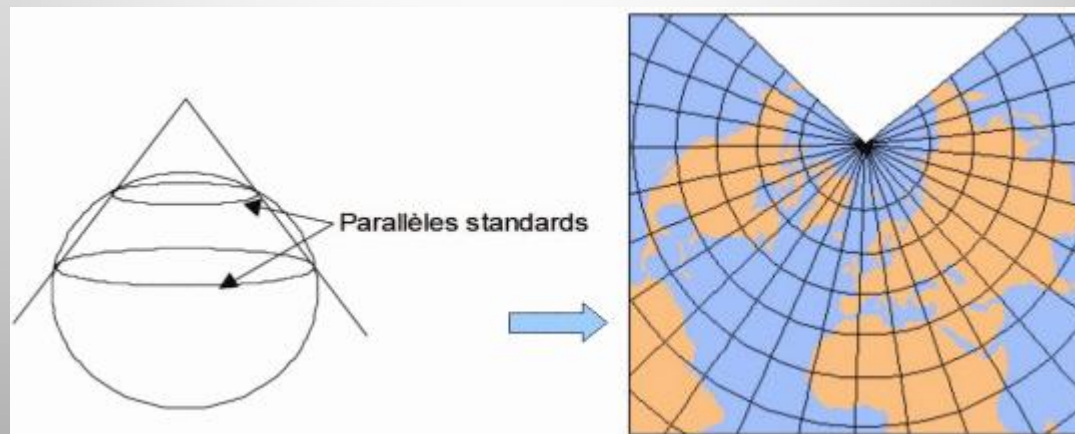


# Systèmes Géographiques et Cartographiques

- Système géographique Français RGF93
  - Ellipsoïde GRS80



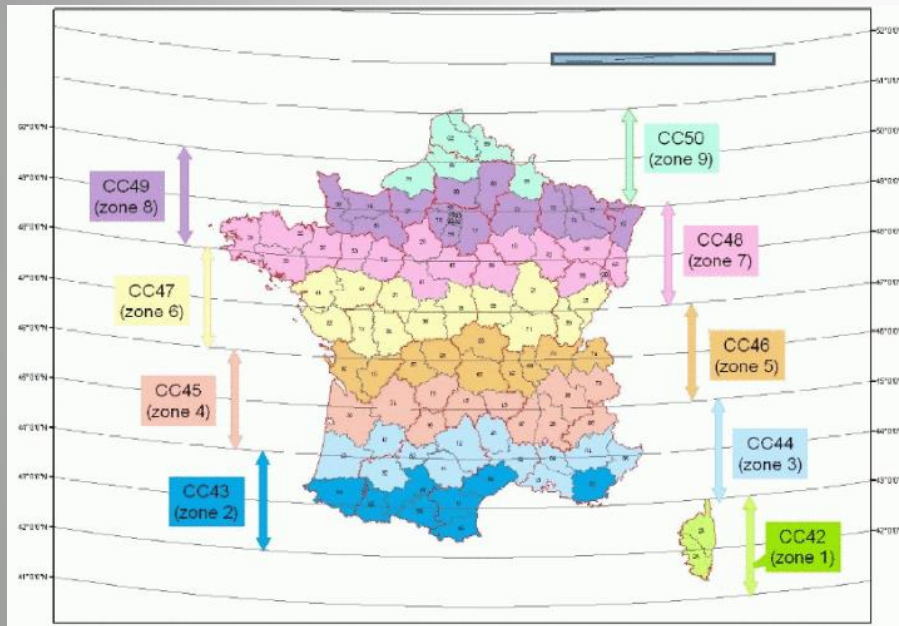
- Projection lambert 93





# Systèmes Géographiques et Cartographiques

- Système géographique Français Lambert CC42...

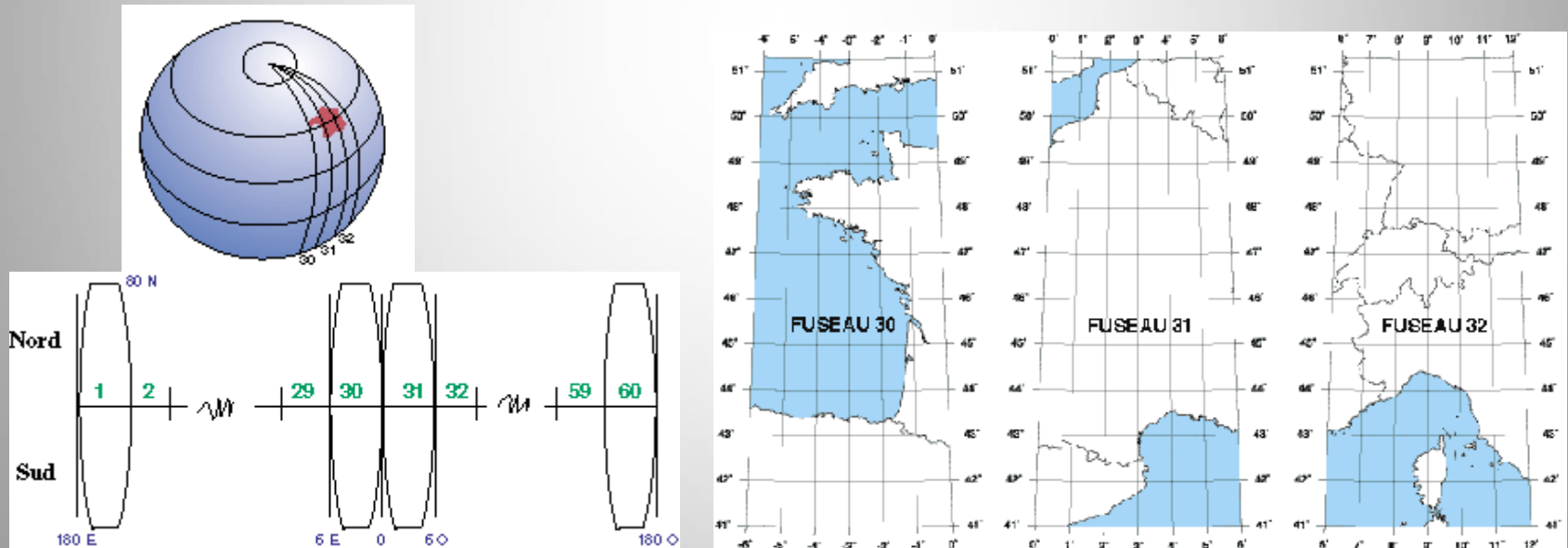


Projection	$\varphi_0$	$\varphi_1$	$\varphi_2$	$X_0$	$Y_0$	EPSG
CC42	42°	41.25°	42.75°	1 700 000 m	1 200 000 m	3942
CC43	43°	42.25°	43.75°	1 700 000 m	2 200 000 m	3943
CC44	44°	43.25°	44.75°	1 700 000 m	3 200 000 m	3944
CC45	45°	44.25°	45.75°	1 700 000 m	4 200 000 m	3945
CC46	46°	45.25°	46.75°	1 700 000 m	5 200 000 m	3946
CC47	47°	46.25°	47.75°	1 700 000 m	6 200 000 m	3947
CC48	48°	47.25°	48.75°	1 700 000 m	7 200 000 m	3948
CC49	49°	48.25°	49.75°	1 700 000 m	8 200 000 m	3949
CC50	50°	49.25°	50.75°	1 700 000 m	9 200 000 m	3950

- 9 projections appelées coniques conformes 9 zones

# Systemes Géographiques et Cartographiques

- GPS: UTM (Universal Transverse Mercator)
  - Système mondial de 122 projections
  - 60 **fuseaux** de  $6^\circ$  (entre  $80^\circ$  Sud et  $80^\circ$  Nord) + 2 poles



- La France: fuseaux UTM Nord 30, 31 et 32

# Systèmes Géographiques et Cartographiques

- Coordonnées GPS: Lat/Lon

- La salle:

- 43.616513, 7.072094 = 43°36'59.5"N+7°04'19.5"E

- Plus d'infos:

- Wikipédia

- IGN: <http://geodesie.ign.fr/index.php> et  
<http://education.ign.fr/dossiers/mesurer-la-terre>

- <http://seig.ensg.eu/>

- [http://sgcaf.free.fr/pages/techniques/ign\\_cooronnees.htm](http://sgcaf.free.fr/pages/techniques/ign_cooronnees.htm)

# Leafletjs

- [leafletjs](https://leafletjs.com/) est une librairie Opensource pour afficher des cartes interactives utiles à la navigation (comme google maps)
- Seulement 33Ko, Tous les browsers
  - Map controls
  - Layers
  - Interaction Features
  - Custom maps



# Geolocalisation sous HTML5

- HTML5: dans le navigator: 92%
- [http://www.w3schools.com/html/html5\\_geolocation.asp](http://www.w3schools.com/html/html5_geolocation.asp)
- Canluse Geolocalisation

Geolocation 📍 - REC

Global

91.56% + 0.01% = 91.57%

Method of informing a website of the user's geographical location

Current aligned

Usage relative

Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			45						
8			46					4.3	
9		43	47					4.4	
10		44	48			8.4		4.4.4	
11	13	45	49	9	35	9.2	8	47	49
	14	46	50	9.1	36	9.3			
		47	51		37				
		48	52						

# Exercices

- <https://github.com/vestri/CoursAR/Course3>
- **geo-sandbox-js-1-basic-map**
  - Affichez votre localisation
  - Tracez le triangle des Bermudes (en rouge)
  - Changer de carte (stamen: <http://maps.stamen.com/> )

# Exercices

- <https://github.com/vestri/CoursAR/Course3>
- **geo-sandbox-js-1-basic-map**
  - Affichez votre localisation
  - Tracez le triangle des Bermudes (en rouge)
  - Changer de carte (stamen: <http://maps.stamen.com/> )
- **geo-sandbox-js-2-current-location**
  - Recentrer carte sur localisation courante
  - Dessiner un cercle avec précision estimée



# Exercices

- <https://github.com/vestri/CoursAR/Course3>
- **geo-sandbox-js-1-basic-map**
  - Affichez votre localisation
  - Tracez le triangle des Bermudes (en rouge)
  - Changer de carte (stamen: <http://maps.stamen.com/> )
- **geo-sandbox-js-2-current-location**
  - Recentrer carte sur localisation courante
  - Dessiner un cercle avec précision estimée
- **geo-sandbox-js-3-poi-distance**
  - Calculez distance à Marseille

# Exercices

- <https://github.com/vestri/CoursAR/Course3>
- **geo-sandbox-js-1-basic-map**
  - Affichez votre localisation
  - Tracez le triangle des Bermudes (en rouge)
  - Changer de carte (stamen: <http://maps.stamen.com/> )
- **geo-sandbox-js-2-current-location**
  - Recentrer carte sur localisation courante
  - Dessiner un cercle avec précision estimée
- **geo-sandbox-js-3-poi-distance**
  - Calculez distance à Marseille  
([https://fr.wikipedia.org/wiki/Distance\\_du\\_grand\\_cercle](https://fr.wikipedia.org/wiki/Distance_du_grand_cercle))

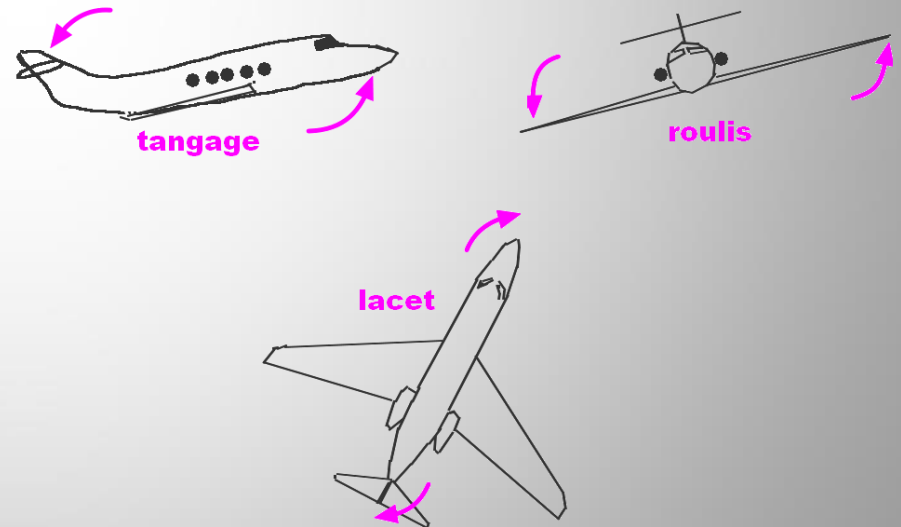
# Capteurs smartphones

- GPS,
- Accéléromètre,
- Gyromètre (confondu avec le Gyroscope)
- Magnétomètre,
- Capteurs de pression,
- Capteurs de lumière ambiante,
- Capteur de proximité.

# Capteurs smartphones

- **six** principaux de degrés de liberté d'un solide dans l'espace:
  - 3 translations:  $t_x$ ,  $t_y$ ,  $t_z$
  - 3 rotations tangage, roulis et lacet

<b><i>D'abord en translation:</i></b>	
• Avant - arrière	<b>X</b>
• Droite - gauche	<b>Y</b>
• Haut - bas	<b>Z</b>
<b><i>Et, en rotation:</i></b>	
• Basculer d'avant en arrière	<b>Tangage</b>
• Basculer de droite à gauche	<b>Roulis</b>
• Pivoter comme les aiguilles d'une montre	<b>Lacet</b>



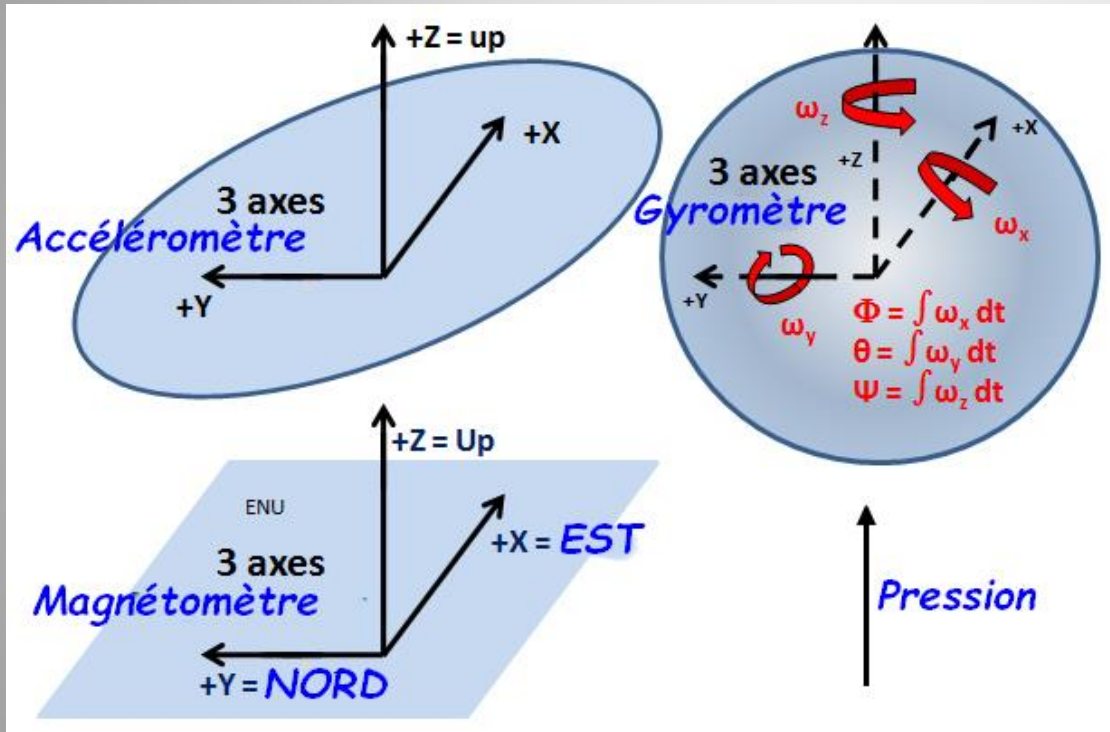
# Capteurs smartphones

- **GPS**
  - Localisation de l'appareil
- **Accéléromètre:**
  - Il ne détecte pas une position, mais une accélération sur chaque X, Y, Z.
  - Permet de savoir dans quelle direction l'appareil se déplace;
- **Gyromètre** (≠ gyroscope qui mesure position angulaire):
  - Le gyromètre ne détecte pas un déplacement linéaire le long d'un axe, mais une accélération de la rotation AUTOUR d'un AXE.
  - Le gyromètre mesure soit des changements dans l'orientation (mouvement angulaires) ou des changements de vitesse de rotation.
- **Magnétomètre**
  - Un magnétomètre mesure les champs magnétiques et parce que la terre possède un champ magnétique significatif,
  - le magnétomètre peut être utilisé comme une boussole.

# Capteurs smartphones

- **Le pedomètre:**
  - Donne le nombre total de pas en 24h, la distance parcourue et l'énergie dépensée.
- **Le capteur d'orientation**
  - Détecte le statut de direction de l'appareil, permet la rotation automatique de l'écran lorsque l'appareil est tourné horizontalement..
- **Le capteur de proximité:**
  - Détecte la présence du corps humain au niveau de l'écouteur de l'appareil.
- **Le Détecteur de luminosité:**
  - Ce capteur permet de savoir quelle est l'intensité lumineuse de l'environnement, ce qui permet de régler automatiquement l'éclairage de l'écran (l'écran consomme beaucoup de courant).

# Capteurs smartphones



C'est donc un système à 10 capteurs d'attitude qui est embarqué

= 3 accéléromètres  
+ 3 gyromètres  
+ 3 magnétomètres  
+ 1 pression



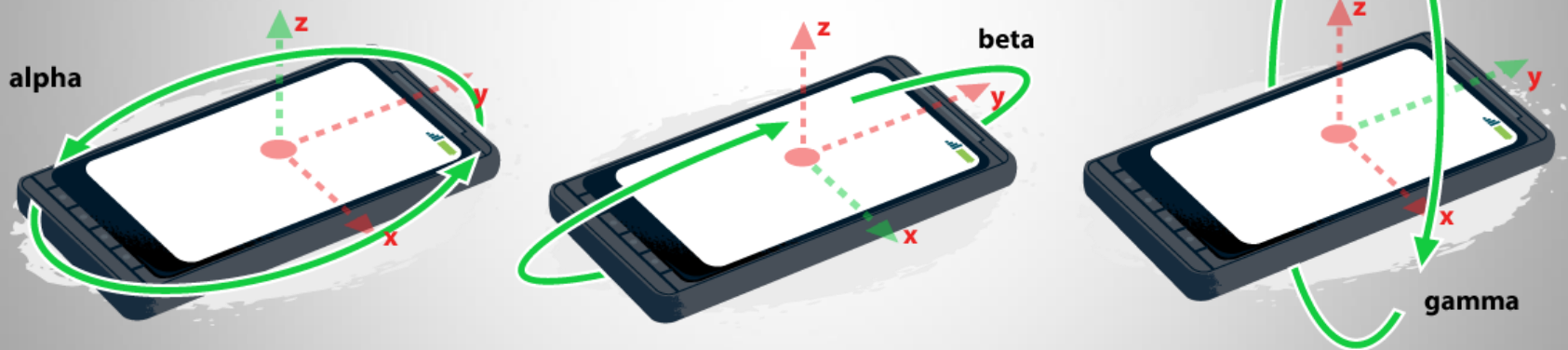
# DeviceOrientation Event Specification

- HTML5: Events définis pour mouse/keyboard...mobile
- <https://www.w3.org/TR/orientation-event/>
- Canluse: DeviceOrientation et DeviceMotion 91%

DeviceOrientation & DeviceMotion events									
Global									
2.07% + 89.34% = 91.41%									
- CR									
API for detecting orientation and motion events from the device running the browser.									
Current aligned Usage relative Date relative Show all									
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android *	Chrome for Android
			49					4.4	
		51	55			9.3		4.4.4	
1 11	14	52	56	10	43	10.2	all	53	56
	15	53	57	10.1	44				
		54	58	TP	45				
		55	59						

# DeviceOrientation Event Specification

- DeviceOrientation:
  - Collecte les données d'inclinaison envoyées par l'accéléromètre
  - L'objet **event** retourne trois propriétés: alpha, beta, gamma

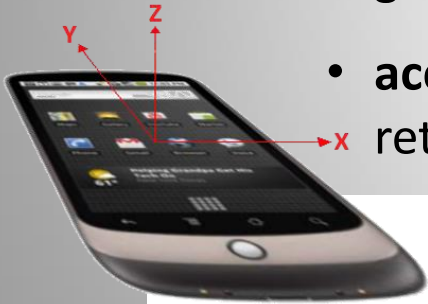


```
if(window.DeviceOrientationEvent) {  
    window.addEventListener("deviceorientation", process, false);  
} else {  
    // Le navigateur ne supporte pas l'événement deviceorientation  
}
```

JavaScript

# DeviceOrientation Event Specification

- DeviceMotion:
  - collecte l'accélération sur les 3 axes ( $\text{m/s}^2$ )
  - L'objet **event** retourne deux propriétés :
    - **acceleration** : L'accélération calculée par l'appareil en enlevant la gravité.
    - **accelerationIncludingGravity** : La valeur de l'accélération brute, retournée par l'accéléromètre.



	Not accelerating	Accelerating up	Accelerating forward	Accelerating right	Accelerating up & to the right
<b>acceleration</b>	{0, 0, 0}	{0, 5, 0}	{0, 0, 2}	{3, 0, 0}	{5, 5, 0}
<b>accelerationIncludingGravity</b>	{0, 9.81, 0}	{0, 14.81, 0}	{0, 9.81, 2}	{3, 9.81, 0}	{5, 14.81, 0}

# Exercices/tests

- Testez DeviceOrientation
- Testez DeviceMotion
- Sur votre smartphone
- <https://www.html5rocks.com/en/tutorials/device/orientation/>

# Pour tester

- Créer un compte sur <https://www.000webhost.com/>
- Ou tout autre free webhosting site
- Uploader vos fichiers
- Tester avec votre smartphone
- Sinon: [www.3dvtech.com/TestSensor/](http://www.3dvtech.com/TestSensor/)

# Exemples Complets

- Testez sensor-sandbox-js-1-log
- Testez sensor-sandbox-js-2-histo

## Github de Nicolas Brignol

- <https://github.com/nbrignol/geo-sandbox-js>
- <https://github.com/nbrignol/sensor-sandbox-js>

# Exercice Final

- **Vuforia (Projet??)**

- D'abord finir l'ancien projet (startwars)

<http://wirebeings.com/star-wars-augmented-reality.html>

- Ajouter un objet géoréférencé/gyroscope

<http://wirebeings.com/markerless-gps-ar.html>

<http://wirebeings.com/markerless-augmented-reality.html>

- **Javascript**

- Mélangez aruco/jsfeat + geoloc ou orientation
- Afficher des objets Geolocalisés flottants



# Rappel

<https://github.com/artmobilis/ArtMobilis-js/wiki/fr-Configuration-framework-nodejs-ionic-android>

- **Chrome:**

- Bloque getUserMedia pour les fichiers locaux
- Lancer avec --disable-web-security pour du debug
- Navigator.getUserMedia plus supporté -> MediaDevices.getUserMedia()
- Il faudrait utiliser adapter.js
- Attention: exemples pas mis à jour -> utilisez Firefox

- **Firefox:**

- Version 40 et +: pb avec les vieilles cartes graphique blacklistées
- Installer version 31 pour du debug (marche sur mon laptop)

# Pour la semaine prochaine

- Récupérez OpenCV

<http://opencv.org/>

- Récupérer les contribs

[https://github.com/Itseez/opencv\\_contrib](https://github.com/Itseez/opencv_contrib)

- Récupérer Cmake

<https://cmake.org/>

- On l'installera puis testera des exemples